

SYSTEMS AND METHODS FOR INFORMATION EXTRACTION

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to United States Provisional Patent Application Serial No. 60/432,130 filed December 9, 2002 entitled "Methods and Apparatus for Information Extraction," which is hereby incorporated in its entirety by this reference.

FIELD OF THE INVENTION

[0002] The present invention relates generally to information extraction and, more particularly, to extracting data from articles.

BACKGROUND OF THE INVENTION

[0003] Many articles, of which documents are one type, are created based on information contained in a database. Such articles have data that remains constant and other variable data obtained from the database. For example, there may be a set of documents each containing a list of items offered for sale. All of these documents contain header data and general sales terms, for example, which would remain constant. The data for each individual item, such as product category, product description, product image, price, and SKU number, is variable and would differ for each individual item. The Internet, for example, contains many web pages similarly

constructed – where several web pages have similar or identical constant data and variable data associated with a variety of information fields.

[0004] It is desirable to extract this unstructured variable data contained in the documents so that the data can be structured. Providing structure to this variable data allows the data to be more easily searched, presented and processed more efficiently than could the original documents. For example, the Internet contains a great number of web sites offering items for sale. However, a user who wishes to purchase a particular item from the Internet conventionally visits the web sites offering the item for sale to compare the sales terms of the item, such as price. Such an approach can be very time consuming. It is, thus, desirable to extract the relevant data about the item from all or several of the websites offering the item for sale and present it to the user in a single web page or a series of web pages.

[0005] In situations such as the Internet, web pages are formatted for viewing by people. Due to the unstructured nature of web pages, the variable data is often embedded with other format and constant data making the identification of variable data difficult. The owner or author of the web pages may not be the entity that desires to extract the data. This results in problems when trying to extract the variable data.

[0006] Some methods and systems exist that attempt to identify, extract and manipulate this variable data. Most of these methods and systems utilize a wrapper procedure. A wrapper uses the formatting conventions of a document to identify the variable data. For example, if one wants to extract price information about items for sale from a document and the item names always appear in bold and the prices always

appear in italics in the document, a wrapper can be created to recognize this format and extract the data identified as bold as the item names and extract the data identified as italics as the prices.

[0007] Some systems and methods use a customized wrapper procedure. Because few document and web page authors publish their formatting conventions, a designer of a customized wrapper must manually construct a wrapper for each type of document. Further, the format of a document may periodically change. As a result, customized wrappers are tedious and error prone.

[0008] Other systems and methods attempt to learn wrappers from labeled examples of the data required to be extracted from documents. One such example is described in Kushmerick, N., Wrapper Induction: Efficiency and Expressiveness, Artificial Intelligence J. 118(1-2):15-68 (2000) (special issue on Intelligent Internet Systems), and <http://www.cs.ucd.ie/staff/nick/home/research/download/kushmerick-aij2000.pdf>. The wrapper induction technique described by Kushmerick involves learning the formatting conventions of a web page from a set of examples of a resource's web pages, each annotated with text fragments to be extracted. The Kushmerick technique typically requires a user to point out examples of the types of fields to extract data from before the wrapper induction is run on a page and the technique has problems properly identifying variable data if the format changes slightly from web page to web page.

SUMMARY OF THE INVENTION

[0009] Embodiments of the present invention comprise systems and methods for information extraction. Embodiments improve identifying information fields in articles and extracting data from articles. One aspect of an embodiment of the present invention comprises receiving a plurality of related articles, determining a seed article from the related articles, identifying at least one information field within the seed article by comparing the seed article to at least one other related article, determining a label for the information field, and associating a pointer to a location of the information field in the seed article to create a template. Comparing the seed article to at least one other related article can be performed by a dynamic programming alignment algorithm to determine an alignment between the seed article and the related article or by other means. Another aspect of an embodiment of the present invention comprises identifying a plurality of templates each comprising at least one information field, comparing the source article to the templates to determine the closest template, and associating data from the article with an information field. Additional aspects of the present invention are directed to computer systems and to computer-readable media having features relating to the foregoing aspects.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] These and other features, aspects, and advantages of the present invention are better understood when the following Detailed Description is read with reference to the accompanying drawings, wherein:

[0011] FIG. 1 illustrates a block diagram of a system in accordance with one embodiment of the present invention;

[0012] FIG. 2 illustrates a flow diagram of a method in accordance with one embodiment of the present invention;

[0013] FIG. 3 illustrates a flow diagram of a method in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

[0014] The present invention comprises methods and systems for information extraction, including methods and systems for identifying and extracting information from articles. Reference will now be made in detail to exemplary embodiments of the invention as illustrated in the text and accompanying drawings. Those skilled in the art will recognize that many other implementations are possible, consistent with the present invention. The same reference numbers are used throughout the drawings and the following description to refer to the same or like parts.

[0015] Various systems in accordance with the present invention may be constructed. FIG. 1 is a diagram illustrating an exemplary system in which exemplary embodiments of the present invention may operate. The present invention may operate, and be embodied in, other systems as well.

[0016] The system 100 shown in FIG. 1 includes multiple client devices 102a-n, a server device 104, and a network 106. The network 106 shown includes the Internet. In other embodiments, other networks, such as an intranet may be used.

Moreover, methods according to the present invention may operate in a single computer. The client devices 102a-n shown each include a computer-readable medium, such as a random access memory (RAM) 108 in the embodiment shown, coupled to a processor 110.

[0017] The processor 110 executes a set of computer-executable program instructions stored in memory 108. Such processors may include a microprocessor, an ASIC, and state machines. Such processors include, or may be in communication with, media, for example computer-readable media, which stores instructions that, when executed by the processor, cause the processor to perform the steps described herein.

[0018] Embodiments of computer-readable media include, but are not limited to, an electronic, optical, magnetic, or other storage or transmission device capable of providing a processor, such as the processor in communication with a touch-sensitive input device, with computer-readable instructions. Other examples of suitable media include, but are not limited to, a floppy disk, CD-ROM, magnetic disk, memory chip, ROM, RAM, an ASIC, a configured processor, all optical media, all magnetic tape or other magnetic media, or any other medium from which a computer processor can read instructions. Also, various other forms of computer-readable media may transmit or carry instructions to a computer, including a router, private or public network, or other transmission device or channel, both wired and wireless. The instructions may comprise code from any computer-programming language, including, for example, C, C++, C#, Visual Basic, Java, and JavaScript.

[0019] Client devices 102a-n may also include a number of external or internal devices such as a mouse, a CD-ROM, a keyboard, a display, or other input or output devices. Examples of client devices 102a-n are personal computers, digital assistants, personal digital assistants, mobile phones, smart phones, pagers, digital tablets, laptop computers, a processor-based device and similar types of systems and devices. In general, a client device 102a-n may be any type of processor-based platform connected to a network 106 and that interacts with one or more application programs. The client devices 102a-n shown include personal computers executing a browser application program such as Internet Explorer™, version 6.0 from Microsoft Corporation, Netscape Navigator™, version 7.1 from Netscape Communications Corporation, and Safari™, version 1.0 from Apple Computer.

[0020] Through the client devices 102a-n, users 112a-n can communicate over the network 106 with each other and with other systems and devices coupled to the network 106. As shown in FIG. 1, a server device 104 is also coupled to the network 106.

[0021] The server device 104 shown includes a server executing an identification and extraction engine application program located in memory 118. Similar to the client devices 102a-n, the server device 104 shown includes a processor 116 coupled to a computer readable memory 118. Server device 104, depicted as a single computer system, may be implemented as a network of computer processors. Examples of a server devices 104 are servers, mainframe computers, networked computers, a processor-based device and similar types of systems and devices. Client

processors 110 and the server processor 116 can be any of a number of well known computer processors, such as processors from Intel Corporation of Santa Clara, California and Motorola Corporation of Schaumburg, Illinois. The server device 104 may also be connected to a database 126.

[0022] The server device 104, or related device, can access the network 106 to locate articles, such as web pages, stored at other devices or systems connected to the network 106. Articles include documents, for example, web pages of various formats, such as HTML, XML, XHTML, Portable Document Format (PDF) files, and word processor, database, and application program document files, audio, video, or any other information of any type whatsoever made available on a network (such as the Internet), a personal computer, or other computing or storage means. The embodiments described herein are described generally in relation to documents, but embodiments may operate on any type of article.

[0023] In one embodiment, the identification and extraction engine 120 identifies information fields in a document or set of related documents and extracts data from a source document using the information fields. The identification and extraction engine 120 includes a template generator 122 and an information extractor 124. In the embodiment shown, each comprises computer code residing in the memory 118. The template generator 122 generates a template identifying information fields in the documents, which contain variable data, for a set of related documents. These information fields are provided a label and a template is generated using the labeled identification fields. In one embodiment, the template generator can

generate templates offline. The information extractor 124 utilizes the generated template to extract data from a source document or documents by comparing the source document with the template and associating the variable data of the source document with the labeled fields. In one embodiment, information extraction can be performed every time a website is crawled, for example, once a week. Other functions and characteristics of the template generator 122 and the information extractor 124 are further described below.

[0024] It should be noted that the present invention may comprise systems having different architecture than that which is shown in FIG. 1. For example, in some systems according to the present invention, the template generator 122 and information extractor 124 may not be part of the identification and extraction engine 120 or may not be located on the same server device. The system 100 shown in FIG. 1 is merely exemplary, and is used to explain the exemplary methods shown in FIGS. 2 – 3.

[0025] Various methods in accordance with the present invention may be carried out. One exemplary method according to the present invention comprises accessing a plurality of related articles, determining a seed article from the related articles and identifying at least one information field within the seed article by comparing the seed article to at least one other related article. The seed article may be compared to at least one other related article, for example, by a dynamic programming alignment algorithm to determine an alignment between the seed article and the related article. The information field may correspond to variable data.

[0026] The exemplary methods may also comprise determining a cluster of related articles from the related articles and this may be performed, for example, by using a dynamic programming alignment algorithm to compute edit distances between the seed article and all of the related articles and choosing the cluster of articles based on the edit distances. At least one information field within the seed article may be identifying by comparing the seed article to the cluster of articles.

[0027] This method may further comprise determining a label for the information field and associating a pointer to a location of the information field in the seed article to create a template.

[0028] In some embodiments, the articles can be web pages and the related articles can be web pages on a web site. In these embodiments, the method may further comprise simplifying the content on a web page and this may include preserving visible text, visible images, and visible paragraph and table formatting.

[0029] Another exemplary method according to the present invention comprises identifying a plurality of templates each comprising at least one information field, comparing the source article to the templates to determine the closest template, and associating data from the article with an information field from the closest template. The source article may be compared to the templates by a dynamic programming alignment algorithm to compute an edit distance between the source article and the templates. The source article may be a web page. The method may further comprise extracting the associated data.

[0030] Fig. 2 illustrates an exemplary method according to an embodiment of the present invention. This exemplary method is provided by way of example, as there are a variety of ways to carry out methods according to the present invention. The method 200 shown in FIG. 2 can be executed or otherwise performed by any of various systems. The method 200 is described below as carried out by the system 100 shown in FIG. 1 by way of example, and various elements of the system 100 are referenced in explaining the example method of Fig. 2.

[0031] The method 200 shown provides a method for identifying data fields within related documents. In one embodiment, the related documents are web pages from a web site, in which the web pages are generated based on a database.

[0032] Each block shown in FIG. 2 represents one or more steps carried out in the exemplary method 200. Referring to FIG. 2, in block 202, the example method 200 begins. Block 202 is followed by block 204, in which a sample set of related documents is obtained. The documents are related in that they share the same or similar template, in which certain data is constant and certain data is variable from document to document. In one embodiment, the template generator 122 crawls all of the web pages of a web site located on a device connected to the network 106 and indexes and caches the web pages. The index and cache could be stored, for example, in the template generator 122 itself, another location in memory 118 or in an external memory location. The template generator then randomly selects web pages from the cache. In another embodiment, the documents may be selected algorithmically or by

inspection. Alternatively, a sample of documents is not selected and all of the relevant documents are processed.

[0033] Next in block 206, in the embodiment shown the documents are processed by the template generator 122 to remove extraneous formatting. For example, in the case of web pages encoded in Hyper-text Markup Language (HTML), the HTML may be simplified so as to reduce the page to a subset of HTML related to layout. In this example, the document is reduced to contain only visible text, images, paragraph and table formatting and all contiguous text is placed on a single line. This is accomplished by removing text formatting, white spaces, JavaScript, comments, forms, etc. Alternative or additional document processing may be carried out.

[0034] In one embodiment, the documents are then compared to create clusters of a number of closely related documents. For example, each sample document may be compared and then aligned with the closest nine documents to create a cluster of ten documents. In one embodiment, the dynamic programming alignment algorithm is used to compare the documents. The dynamic programming alignment algorithm compares and aligns documents to compute relative scores for the compared documents known as edit distances. The edit distance can be a number, roughly proportional to the number of insertions and deletions necessary to transform one document into another. An alignment of two documents, for example, can be a list of those insertions or deletions, or equivalently, a mapping from parts of one document to parts of another. Dynamic programming alignment is a method understood by those skilled in the art, and accordingly need not be described in further detail herein.

Further details on dynamic programming may be found in Dan Gusfield, Algorithms on Strings, Trees, and Sequences, Computer Science and Computation Biology (Cambridge University Press 1997), which is incorporated herein by reference.

[0035] Then in block 208, a “seed” document is selected. A seed document is generally the document that indicates where constant data appears and where variable data appears in the sample documents. Preferably, the seed document is the best document in the available sample set of documents to so indicate. In one embodiment, the seed document is automatically chosen by the template generator 122. In the embodiment explained above, the template generator 122 organizes the documents into clusters of closely related documents. The template generator 122 then selects a document from the cluster as the seed document for the cluster. The seed document could be the document with the smallest edit distances to the other documents in the cluster.

[0036] For example, for a website that sells CDs, books and toys, all of the website’s web pages for CDs will generally be formatted the same, as will the web pages for books and toys. The web pages for CDs will generally differ in format from the web pages for books and toys. The template generator would first crawl the website and select a sample set of web pages. When the template generator aligns the documents to form related clusters of documents, at least three clusters of documents, for example, would be formed – one for CD web pages, one for book web pages and one for toy web pages. The template generator would then select a web page from each of the three clusters to be used as a seed document for that cluster.

[0037] In another embodiment, the sample documents are presented by the server device 104 to a user 112a at client 102a, who selects the seed document. In this embodiment, the user 112a would select the seed document based on a determination of the relative suitability of the various documents as a template. In the example above, the template generator may present all of the clusters to the user. The user would then select a seed document for each of the clusters or, for example, only for the CD web pages if the user is interested in only obtaining CD information. The seed document could be selected in a variety of other ways known to those skilled in the art. For example, the seed document may be provided by the author or creator of the documents.

[0038] Next, the seed document identified in block 208 is compared to the sample documents in block 210 by the template generator 122 to determine the similarity between the compared documents. In one embodiment, the seed document is compared to its cluster of most closely related documents using a dynamic programming algorithm to align the seed document with the cluster documents. Alternatively, the seed document can be aligned with all of the sample documents.

[0039] For example, if the documents are web pages containing information about CDs, each web page could contain constant data such as the terms "ARTIST", "TITLE", "DESCRIPTION", and "PRICE" and each web page would contain the variable data associated with each term. The two web pages are then aligned to form pairs of data so that the same constant terms and variable data in each page match up. In one embodiment, the seed document is aligned with all of the cluster documents at

once. In another embodiment, a star alignment algorithm is used to approximate the alignment of the seed document with the cluster documents. The star alignment algorithm is known to those skilled in the art.

[0040] Next in block 212, the variable data portions in the documents are determined by the template generator 122 based on the comparisons performed in block 210. For example, in one embodiment, the alignments of the each of the cluster documents with the seed document are analyzed to determine the constant data portions of the seed document and the variable data portions of the seed document. In another embodiment, the alignments of all of the sample documents with the seed document are analyzed. Where the data on the documents differs from the data on the seed document indicates variable data and where the data on the documents is the same as the data on the seed document indicates constant data. In the example given above, after alignment of the web pages, where there are pairs of identical data, such as, PRICE-PRICE, this indicates constant data and where there are pairs of dissimilar data, such as, \$12.98-\$13.98, this indicates variable data.

[0041] It should be understood that "variable" and "constant" need not be determined using absolute criteria. "Variable" is an indication of a relatively high degree of variability, and/or a predetermined (e.g., absolute) degree of variability. "Constant" is an indication of a relatively high degree of constancy, and/or a predetermined (e.g., absolute) degree of constancy. For example, in one embodiment, it could be required that six out of ten documents should differ from the seed document in a particular location before a field is declared variable.

[0042] In block 214, the variable portions of the documents identified in block 212 are identified as information fields and labeled. In one embodiment, to facilitate identification of information fields, a list of the variable portions of the seed document are presented to a user 112a by the server device 104, grouped by the part of the seed document to which they align. The identified information fields of relevance are labeled by the user 112a for referencing. In the example given above, the user would receive identified information fields containing artist, title, description and price data for all of the cluster documents and would label them as such through the use of a user interface, for example. In one embodiment, the user may select from a predefined list of field names or enter the field name of their choice. The user may be presented with certain information fields that the user is not interested in and will not label.

[0043] Alternatively, the template generator automatically generates labels for relevant information fields. In this embodiment, the template generator could utilize machine learning techniques known to those skilled in the art in conjunction with variable data previously extracted from other documents to label the information fields. Each label for the identified information fields is associated with a pointer to the location of the corresponding variable portion in the seed document to create a template.

[0044] The template can then be used to extract variable data from documents as is described, for example, with reference to FIG. 3 below. Multiple templates may be created for each document type or format. For example, a web site may have

multiple web page formats, and a template corresponding to each different web page format may be created. Those skilled in the art will also recognize that the functions described with reference to each block in FIG. 2 are illustrative only, and are not intended to be limiting.

[0045] FIG. 3 illustrates an exemplary method according to an embodiment of the present invention. This exemplary method is provided by way of example, as there are a variety of ways to carry out methods according to the present invention. The method 300 shown in FIG. 3 can be executed or otherwise performed by any of various systems. The method 300 is described below as carried out by the system 100 shown in FIG. 1 by way of example, and various elements of the system 100 are referenced in explaining the example method of Fig. 3.

[0046] The method 300 shown provides a method for extracting data from a source document or documents. In one embodiment, the source document is a web page on a web site, in which the web page is generated based on a database.

[0047] Each block shown in FIG. 3 represents one or more steps carried out in the exemplary method 300. Referring to FIG. 3, in block 302, the example method 300 begins. Block 302 is followed by block 304, in which one or more templates are identified by the information extractor 124. The templates may, for example, be the templates created previously with respect to Fig. 2 and be all of the templates associated with a particular web site. Next in block 306, the source document is compared with the templates by the information extractor 124 to determine the closest template. In one embodiment, the source document is aligned with each of the

identified templates utilizing dynamic programming alignment techniques and the closest template is determined using edit distances computed for each template as aligned with the source document.

[0048] If for example, the source document is a web page listing CDs for sale from a website that sells CDs, books and toys, the template generator may have three templates, one for CD web pages, one for book web pages and one for toy web pages. The information extractor 124 would compare the source web page to the three templates utilizing the dynamic programming alignment algorithm to compute pairwise edit distances for the three templates in relation to the source web page. The CD template would likely have the smallest edit distance and be selected.

[0049] In block 308, data within the source document is then associated with the information fields labeled in the template by the information extractor 124. For example, once the template and source document are aligned, pointers in the template are used to point to information fields containing variable data in the source document. In the example discussed above, the pointers on the CD template are used to point to the location on the web page that contains the artist, title, description and price variable data. In block 310, the data within the source document corresponding to each information field is extracted by the information extractor and labeled with the corresponding information field label.

[0050] The extracted data can then be stored in a database where it can be searched, presented, and processed more efficiently than could the original documents. In the example above, the artist, title, description and price variable data

is labeled as such and stored in a database with an indication of the source website. This can be done for multiple website and all of the information can be presented to a user when the user selects to view all available CDs. Alternatively, the user can search the information in the database by the variable data, for example, artist, title, description, and price.

[0051] While the above description contains many specifics, these specifics should not be construed as limitations on the scope of the invention, but merely as exemplifications of the disclosed embodiments. Those skilled in the art will envision many other possible variations that are within the spirit and scope of the invention as described above and defined in the following claims.